patent, and further in view of U.S. Patent No. 6,151,704 ('704 Radigan). These rejections are respectfully traversed.

Applicant has disclosed exemplary embodiments for modifying serial dependencies in a procedure. Included is an exemplary second memory operation which is moved to a new position in a graph representation that is closer to a first memory operation. The step of moving the second memory operation includes (i) removing one or more of serial dependencies in initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation. As exemplified in FIG. 10A, it is possible to remove a dependency edge from 1006 to 1008 and replace it with a new dependency edge from 1002 to 1008. A dashed arrow from 1002 to 1008 represents this new dependency edge. Upon removal of the edge, or dependency, from 1006 to 1008, a compiler 58 can move load 1008 out of the repetitive loop, as shown in FIG. 10B (e.g., specification at page 20, lines 25-33). Such a movement of memory operation as shown in FIG. 10B can provide enhanced and efficient machine code compared to the prior depiction of FIG. 10A because load 1008 is executed only once in FIG. 10B rather than on a repetitive basis as in FIG. 10A (e.g., specification at page 21, lines 1-3). The exemplary method for modifying serial dependencies in a procedure uniquely defines an appropriate dataflow algorithm capable of moving data dependencies out of loops to provide enhanced and efficient machine code.

The foregoing features are broadly encompassed by claim 1 which recites, among other features, a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph

representation that is closer to a first memory operation, wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation.

The documents relied upon by the Examiner, regardless of whether they are considered alone or in the combination discussed by the Examiner, fail to teach or suggest features of claim 1. The documents relied upon by the Examiner fail to teach or suggest a method for modifying serial dependencies in a procedure which defines an appropriate dataflow algorithm to move data dependencies out of loops.

The Examiner admits at page 5 of the Office Action that "Moreno does not expressly disclose the limitations wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between said first memory operation and said second memory operation." The Moreno et al. patent teaches inserting extra code to perform an "interference" test, but the Moreno et al. patent does not teach or suggest removing a serial dependency in an initial set of serial dependencies. The Moreno et al. patent does not teach or suggest a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph representation that is closer to a first memory operation, wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii)

creating a new serial dependency between the first memory operation and the second memory operation, as recited in claim 1.

The Muthukumar et al. patent does not cure the deficiencies of the Moreno et al. patent. The Muthukumar et al. patent discloses testing of serial dependencies by facilitating speculative substitution for comparison of execution results. As shown in FIG. 3, the Muthukumar et al. patent discloses a method 300 for implementing compare speculation in software-pipelined loops by modifying a data dependence graph to enable compare speculation. Specifically, the Muthukumar et al. patent discloses replacing a loop-carried-edge 310 from the data dependence graph with a speculative loop-carried edge 320 for speculative (test) execution and comparison (col. 4, lines 37-44). However, the control speculative execution as taught by the Muthukumar et al. patent does not teach or suggest a method for modifying serial dependencies in a procedure to move a second memory operation to a new position in a graph representation that is closer to a first memory operation, wherein the moving step includes removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation. The Muthukumar et al. patent fails to teach or suggest a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph representation that is closer to a first memory operation, wherein the moving step includes (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation, as recited in claim 1.

The Bharadwaj patent does not cure the deficiencies of the Moreno et al. patent and the Muthukumar et al. patent. The Bharadwaj patent discloses that dependency path vectors can be used to "determine if particular code motion or code reordering may be performed without altering the meaning of the source code" (col. 6, lines 41-43). However, the Bharadwaj patent does not teach or suggest a method for modifying serial dependencies in a procedure to move a second memory operation to a new position in a graph representation that is closer to a first memory operation.

The Ruf patent, the '398 Radigan patent, and the '704 Radigan patent do not cure the deficiencies of the Moreno et al. patent, the Muthukumar et al. patent and the Bharadwaj patent, even when considered in the combination relied upon by the Examiner. The Ruf patent was cited for its disclosure of a dependence analysis module 314 and a partitioning algorithm module 316 (col. 9, line 66 to col. 10, line 21); the '398 Radigan patent was cited for its disclosure of an intermediate language representation (e.g., col. 6, lines 56-61); and the '704 Radigan patent was cited for its disclosure of optimizing a loop in a computer program (e.g., col. 6, lines 3-11). The Ruf patent, the '398 Radigan patent, and the '704 Radigan patent do not teach or suggest at least the moving step, including removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and creating a new serial dependency between the first memory operation and the second memory operation.

Even if the references could have been combined in the manner asserted by the Examiner, the combination would not have resulted in the claimed method for modifying serial dependencies in a procedure. For example, combining of features

from the Moreno et al. patent, the Muthukumar et al. patent, the Bharadwaj patent and the Ruf patent, in the manner asserted by the Examiner would not have resulted in a method for modifying serial dependencies in a procedure, including moving a second memory operation to a new position in a graph representation that is closer to a first memory operation. None of the references relied upon by the Examiner teach or suggest (i) removing one or more of serial dependencies in an initial set of serial dependencies that is associated with the second memory operation and (ii) creating a new serial dependency between the first memory operation and the second memory operation, as recited in claim 1.

For the foregoing reasons, Applicant's claim 1 is allowable. Independent claims 13 and 25 recite features similar to those discussed above, and are therefore also allowable. The remaining dependent claims recite additional advantageous features which further distinguish over the documents relied upon by the Examiner. As such, the present application is in condition for allowance.

All objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the application is in condition for allowance and a Notice of Allowance is respectfully solicited.

Respectfully submitted,

BUCHANAN INGERSOLL PC

Date: January 25, 2006

By: [signature] for Reg. No. 48,360
Patrick C. Keane
Registration No. 32,858

P.O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620